

ANÁLISIS DE REQUERIMIENTOS PARA EL DESARROLLO DE SISTEMAS MULTI AGENTES DE TIEMPO REAL¹

REQUIREMENT'S ANALYSIS FOR DEVELOPING REAL-TIME MULTI AGENT SYSTEMS

LUIS E. CASTELLANO²

RESUMEN

Se realiza una revisión de los aspectos teóricos que describen y caracterizan a la tecnología de agentes software, los sistemas multi agentes y los sistemas de tiempo real. A partir de tal caracterización y con base en las potencialidades que representan para la ingeniería del software se analizan los elementos a considerar para el desarrollo de Arquitecturas de Sistemas Multi Agentes de Tiempo Real, en vista de lo cual se resalta la necesidad de concentrarse en la construcción de Arquitecturas de Sistemas Multi Agentes de Tiempo Real Cooperativos para lo que se propone existencia de un Agente Negociador de Tiempo Real y, para algunos casos específicos, un Agente Planificador de tareas y recursos en aras de satisfacer las restricciones de tiempo que imponen los sistemas de tiempo real.

Palabras Claves: Agentes Software, Sistemas Multi Agentes, Sistemas de Tiempo Real, Arquitecturas Software.

ABSTRACT

A review is made on the theoretical aspects that describe and characterize the software agent technology, multi agent systems and real-time systems. From this characterization, and based on the potential they represent for the software engineering, the elements to be considered for the development of Real-Time Multi Agent System Architectures are analyzed, in view of which, the need to concentrate on the build of Cooperative Real-Time Multi Agent System Architecture is highlighted, followed for the proposal of the existence of a Real-Time Agent Negotiator, and for some specific cases, a Scheduler Agent for tasks and resources in order to satisfy the time constraints imposed by real-time systems.

Key Words: Software Agents, Multi Agent Systems, Real-Time Systems, Software Architectures.

1.- INTRODUCCIÓN

El surgimiento de la tecnología de agentes software, ha representado una verdadera revolución en el campo de la Ingeniería del Software, al presentar ésta la posibilidad de avanzar desde, por ejemplo, el desarrollo de aplicaciones con capacidad de acción limitada por un guión de trabajo previamente establecido por el programador y altamente dependiente de las acciones que ejecuten elementos del ambiente (especialmente el usuario) así como de los recursos y capacidades técnicas, de acceso a información u otros disponibles en la plataforma en la que se ejecuta, hacia una visión de un agente, que tal como su nombre lo indica, actúa en nombre de un usuario con ciertas instrucciones para lograr una tarea dada, pero con capacidad de tomar decisiones que le permiten cierta "libertad de acción", por lo que pueden escoger los mecanismos más adecuados para realizar la tarea que le ha sido asignada, tomando

en cuenta las necesidades que deben satisfacerse para lograr la misma.

Desde el punto de vista formal, se establece que un agente software ha de tener la habilidad de llevar a cabo ciertas acciones de manera autónoma para lograr un determinado objetivo en nombre de un tercero (Piszc, 1988). Este ideal de autonomía implica que los agentes poseen la capacidad de interactuar con otras entidades humanas, máquinas u otros agentes software para requerir información o recursos, movilizarse hacia otras plataformas o ambientes que le garanticen los recursos actuando como una entidad dinámica, en el cual no solamente operará sin necesidad de intervención externa sino que será proactivo con capacidad de adaptarse a los cambios en esos varios ambientes y/o plataformas, entre otras capacidades destacables.

Sin embargo, a pesar del creciente interés que este nuevo paradigma ha generado, aun existen ciertas limi-

1 Recibido: Julio 2011

Aprobado: Diciembre 2011

2 Programa Ingeniería, Arquitectura y Tecnología, Vicerrectorado de Planificación y Desarrollo Social, UNELLEZ.
Email: lcastella@gmail.com

taciones en la operacionalización de los mismos de tal manera que se satisfagan los atributos que los definen. Algunas de las limitaciones más relevantes que se pueden mencionar son:

1. *Las dificultades para precisar una definición clara:* En el ámbito académico existe una gran dificultad y polémica a la hora de definir lo que es un agente software y sus características fundamentales. Esto quizás debido a que originalmente la idea de agentes surge del área de la Inteligencia Artificial y luego es trasladada hacia la ingeniería software y los sistemas autónomos, cada una de ellas con su propia visión y por supuesto con sus propias dificultades a la hora de distinguirlos de una simple pieza de software.
2. *Las barreras de la comunicación entre agentes:* Uno de los aspectos que distinguen a un agente es su capacidad de comunicarse con otras entidades para el logro de sus metas particulares. Ahora bien, existen barreras que aún deben ser resueltas para que esta comunicación pueda darse sin afectar la autonomía que cada agente debe poseer, tomando en cuenta restricciones de prioridades de acceso a recursos, planificación según necesidades, etc. Algo que se hace aún más difícil cuando los agentes que están interactuando son desarrollados sobre una plataforma donde, por sus características, no es posible interactuar con otro agente desarrollado en otra plataforma, dadas diferencias hasta en algunos casos casi insalvables en cuanto a ontologías, elementos de vocabulario y algunas otras particularidades de lenguaje, entre otras.
3. *La operacionalización de una Arquitectura Estándar:* Una arquitectura software es un patrón que define la forma en que habrá de organizarse determinado tipo de sistemas y que comprende una descripción de los componentes del mismo, sus propiedades y la forma como se organizan. En este sentido, gran cantidad de arquitecturas para el desarrollo de sistemas Multi Agentes han sido desarrolladas. En estas se describen los elementos a tomar en cuenta para la constitución de sistemas Multi Agentes tal como fue descrito en el punto anterior, sin embargo, las mismas son diseñadas bajo plataformas y/o protocolos específicos y, aunque en general logran funcionar adecuadamente, aun presentan la imposibilidad de comunicarse con otros agentes desarrollados bajo arquitecturas que corresponden a otras plataformas, lo cual presenta nuevamente una limitante para la constitución de Sistemas Multi Agentes.
4. *La creciente necesidad de integración con otras áreas:* Tal como se afirmó al principio, los agentes software se han erigido como un área de creciente

interés dado que introducen elementos que permiten avanzar hacia el desarrollo de aplicaciones más complejas, lo que implica una ampliación de los límites de la Ingeniería del Software. En tal sentido, si se asume que la Tecnología de Agentes está ampliando los límites de la Ingeniería del Software, habría que asumirse que las diferentes áreas de desarrollo de esta (Sistemas distribuidos, arquitecturas, sistemas de tiempo real, componentes, patrones de diseño, recuperación de información, por mencionar algunas) tendrían que poder ofrecer nuevas posibilidades de desarrollo de aplicaciones, ampliación de las capacidades de las ya existentes o la introducción de innovaciones que en definitiva ofrezcan una ampliación del horizonte de posibilidades del área de conocimiento y los problemas que esta resuelva. En el caso específico de los agentes software ha habido grandes avances en este sentido, no en todas las áreas pero sí en algunas bastante significativas, como es el caso de la recuperación de información y los sistemas distribuidos, que sin embargo aun se mantienen con serias deficiencias, no solo por las dificultades habituales que surgen al integrar dos tecnologías, sino también por las limitaciones mismas que de por sí existen en el área de los agentes software y que han sido esbozadas anteriormente.

A través de este trabajo se pretende profundizar sobre este último aspecto, es decir, la integración de los sistemas agentes con otras áreas de la Ingeniería de Software y las potencialidades de crecimiento que representa para esta disciplina, al pretender establecer los requisitos que deben satisfacerse para el desarrollo de Arquitecturas de Sistemas Multi – Agentes de Tiempo Real, especificando las potencialidades que ello podría representar para la Ingeniería del Software.

Se toma en cuenta a los sistemas de tiempo real por ser este un mecanismo que presenta restricciones de tiempo para la ejecución de tareas a un sistema, con lo que se asume que su integración con los sistemas agentes les agrega a los mismos, elementos que permitirían desarrollar nuevas aplicaciones que ampliarían el espectro de acción de la ingeniería de software.

Con la finalidad de abordar todos estos elementos, se ha desarrollado un estudio teórico a través de una investigación pura de tipo documental basada en un diseño de tipo bibliográfico en concordancia por lo establecido por la Universidad Santa María [USM] (2000), donde en un principio se revisará lo alcanzado hasta ahora desde el punto de vista conceptual para definir los agentes software. Así mismo, a partir de tales definiciones se resaltan los atributos que caracterizan a un agente y se hará una revisión de las clasificaciones de agentes hasta ahora propuestas. En una segunda parte, se definen y caracterizan los Sistemas Multi Agen-

tes, los mecanismos de comunicación de agentes, así como las arquitecturas propuestas, los intentos de estandarización y sus plataformas. A continuación se describirán las características y elementos presentes en los Sistemas de Tiempo Real, así como el “estado del arte” relativo al desarrollo de arquitecturas de Sistemas Multi Agente en Tiempo Real, sus problemas y limitaciones. Finalmente se realizará una propuesta donde se evaluarán los elementos básicos de una arquitectura de Sistemas Multi – Agentes y que habría que tomarse en cuenta para incorporar a la misma, restricciones de Tiempo – Real, evaluando las posibilidades de integración de todo ello.

Como resultado final se ofrece una definición de los elementos necesarios para la construcción de una arquitectura de sistemas multi – agentes con restricciones de tiempo – real. El eventual desarrollo de tal arquitectura permitiría la ampliación del marco de posibilidades de desarrollo de aplicaciones informáticas en áreas como las de control de procesos, monitoreo de plantas nucleares, producción y manufacturas, sistemas de control satelital de vehículos, control de tráfico aéreo, telecomunicaciones, multimedia, telemedicina, monitoreo de terapia intensiva y sistemas de defensa, comercio electrónico, robótica, búsqueda y recuperación de información en la web; lo cual implicaría no solo un avance sustancial para el desarrollo de software, sino también para áreas vitales de la sociedad moderna.

2.- AGENTES SOFTWARE

Un agente, cualquiera sea su naturaleza, es una entidad que ejecuta una tarea en nombre de un tercero y según sus requerimientos, pero con la potestad de decidir la forma que considere mas adecuada para lograr tal tarea, usando su conocimiento y experiencia en el área, capacidad de acceso a la información y de asociación con otros agentes y/o proveedores de servicios, etc.

Ahora bien, se podría suponer que un agente software es una pieza de software que de alguna manera puede actuar como un agente según lo descrito en el párrafo anterior. Sin embargo, a pesar del tiempo transcurrido desde el surgimiento de la tecnología de agentes software, una conceptualización de los mismos es aún un tema difícil y objeto de controversias entre los autores. En este sentido Shoham (1999) afirma que tener una conversación sobre agentes software no es fácil dado que nadie puede dar una clara y comprensiva definición de lo que significa y en el mejor de los casos lo máximo que se puede obtener es una definición de una versión particular sobre el concepto, que excluye varios elementos que otras versiones reclamarían como la esencia de los agentes software, lo que se puede observar claramente a través de la revisión de las definiciones

elaboradas por algunos autores y que se presentan a continuación:

1. Jennings & Wooldrige (1998): Un agente es “un programa autocontenido capaz de controlar su proceso de toma de decisiones y de actuar basado en la percepción de su ambiente, en persecución de uno o varios objetivos”
2. Nwana (1996): “Definimos un agente como un componente software y/o hardware que es capaz de actuar exigentemente en orden de conseguir tareas en nombre de sus usuarios”
3. Maes (1994): “Los agentes autónomos son sistemas computacionales que habitan en algún ambiente dinámico y complejo, percibiendo y actuando autónomamente en este ambiente, y al hacer esto llevan a cabo una serie de objetivos o tareas para los cuales fueron diseñados”
4. Smith, Cypher & Spohrer (1994): “Definimos a un agente como una entidad de software persistente dedicada a un propósito específico. ‘Persistente’ distingue a los agentes de las subrutinas; los agentes tienen sus propias ideas sobre como ejecutar las tareas, sus propias agendas. ‘Propósito especial’ los distingue de las aplicaciones multifuncionales completas; típicamente los agentes son mucho más pequeños”

Tal como puede verse, las respuestas a una definición de agentes software tienden a ser tan generales e imprecisas como variadas, en este sentido los investigadores han intentado distinguir las características o atributos de manera que se pueda tener un marco de referencia que permita identificar a los agentes software (Piszcz, 1988; Shoham, 1999; Jennings & Wooldrige, 1998; Nwana, 1996; Maes, 1994; Frankling & Graesner, 1996; Russell & Norvig, 1995; Amandi, 2001; Smith, Cypher & Spohrer, 1994; Hayes-Roth, 1995; Griss, 2000; Tolosa & Bordignon, 1999; Agent Working Group [AWG], 2000; Jennings, Sycara & Wooldridge, 1998):

1. **Autonomía:** Es capaz de actuar por si mismo, sin necesidad de intervención, control o supervisión humana o de otras entidades.
2. **Adaptabilidad:** Tienen la capacidad de adaptarse a las exigencias, comportamientos y preferencias de los usuarios así como a los cambios en el ambiente, a través del tiempo de acuerdo a sus experiencias.
3. **Colaboración:** pueden colaborar con otros agentes o entidades para compartir información o para exponer los resultados de la tarea asignada si el logro de sus objetivos así lo requiere o para coordinarse para lograr un propósito común.

4. **Sociabilidad:** debe tener habilidad para interactuar con otros agentes o incluso con alguna persona, si es necesario para cumplir con las tareas que se le han asignado.
5. **Coordinación:** Son capaces de ejecutar algunas actividades en ambientes compartidos por otros agentes.
6. **Movilidad:** El accionar de un agente no está limitado al sistema en el cual se está ejecutando. Este tiene la habilidad particular de transportarse a sí mismo desde un sistema en una red particular hacia otro en función de conseguir recursos para lograr su tarea o continuar con su ejecución, de ser necesario, si el sistema en donde se encuentra no le puede asegurar tales cosas.
7. **Reactividad:** Un agente debe poder reconocer el estado de su ambiente (que puede ser el mundo físico, una interfaz gráfica de usuario, una aplicación, un conjunto de otros agentes o quizás una combinación de algunos o todos los anteriores) y, en función de esto, actuar respondiendo de manera adecuada a los cambios producidos en el mismo.
8. **Orientación por Objetivos:** Un agente no solo debe actuar por cambios detectados en el medio ambiente, sino que además debe trabajar en función de los objetivos para los cuales fue diseñado y las tareas que le fueron delegadas en cada momento.
9. **Continuidad Temporal:** A diferencia de otras rutinas y programas que son ejecutados para el logro de tareas particulares y luego desaparece, los agentes se ejecutan de manera continua por largos períodos de tiempo hasta que se hayan alcanzado sus objetivos o hasta que su ciclo perdure y el usuario no desee detenerlo.
10. **Proactividad:** El agente no solo es capaz de reaccionar a los cambios en el ambiente, sino que puede tomar la iniciativa en función de lograr sus objetivos.
11. **Racionalidad:** tiene la capacidad de reconocer y ejecutar determinadas acciones en base a sus metas internas a sabiendas que tales acciones en particular le permitirán acercarse al logro de sus objetivos, de acuerdo a sus creencias y conocimientos.
12. **Veracidad:** Se asume que un agente es incapaz de proporcionar falsa información de manera consciente.
13. **Competitividad:** su capacidad de coordinación con otros agentes exceptúa el caso en el que el logro del objetivo de uno de ellos implica la falla de todos los demás.

Aunque la mayoría coincide en que se puede considerar la presencia de un agente cuando este es capaz de mostrar algún grado de movilidad, inteligencia, autonomía, colaboración, adaptabilidad, persistencia o personalidad; no todos los autores están de acuerdo en que la totalidad de las características nombradas con anterioridad han de ser las propiedades fundamentales que definen a un agente llegando al plantear a algunas de estas como propiedades complementarias o explicativas de otras más fundamentales o en algunos casos afirmando que no son atributos propios de un agente, dado que son propiedades que surgen del tipo de acciones que este desarrolla para lograr sus tareas, por ello se han elaborado algunas clasificaciones de agentes y que serán presentadas a continuación.

2.3.- Clasificación:

Existen diversas clasificaciones que dan cuenta de los diferentes tipos de agentes "identificables", sin embargo, estas clasificaciones no han estado exentas de polémicas en cuanto a los criterios utilizados para su elaboración. En algunos casos se hace énfasis en el objetivo o el tipo de tareas que ejecutan mientras que en otros se enfatiza en la tecnología que soporta al agente o en algún atributo que destaca y, aunque se ha pensado en una clasificación bidimensional, esta idea no es totalmente aceptada dado que existen casos en los que, dado el objetivo de un agente, se debe destacar algún atributo o acarrear una tecnología particular para el logro de este, además de que algunos consideran que se podría incluir en una clasificación los tipos de estructuras de control que utilizan, los tipos de ambientes en los que se desenvuelven o lenguajes en que son escritos, solo por mencionar algunas opiniones. En todo caso parece haber un acuerdo en que habría que esperar por un mayor desarrollo de la tecnología de agentes software para una evaluación más profunda en esta área.

Sin intención de juzgar acerca de los criterios utilizados, a continuación se listan algunos de los tipos de agentes identificados hasta ahora (Nwana, 1996; Tolsa & Bordignon, 1999; AWG, 2000):

1. **Agentes de Interfase:** Un agente de interfase es un programa que es capaz de operar como una interfase de usuario y asistir a este en la operación de la misma y la manipulación de las operaciones del sistema detrás de la interfase.
2. **Agentes Móviles:** Son procesos de software que tienen la capacidad de trasladarse a través de diversas redes de computadores interactuando con diferentes servidores recopilando información en nombre del usuario para completar la tarea encomendada por este y regresar al sitio de origen.

3. **Agentes de Recuperación de Información:** Los agentes de recuperación han sido propuestos como una de las soluciones posibles que permiten asistir a los usuarios en la formulación de consultas avanzadas en la Web. Pueden tener mecanismos de búsqueda y navegación flexibles y algoritmos de clasificación poderosos.
4. **Agentes Adaptativos:** Son aquellos capaces de tener algún grado de respuesta a otros agentes y su ambiente. es decir, son capaces de reaccionar con una respuesta previamente determinada ante un evento o señal particular en el ambiente, como por ejemplo los que manejan sensores de calor o energía.
5. **Agentes Coordinativos:** Los agentes coordinativos se constituyen como partes de un sistema de agentes que intenta recrear de manera análoga el funcionamiento de diferentes tipos de organizaciones humanas, siendo estos sistemas compuestos por varios tipos de agentes que reproducen, cada uno de ellos, funciones específicas de cada tipo de organización. Por ejemplo, en una empresa de distribución de productos existen diferentes áreas encargadas de las compras, solicitudes de productos por parte de los clientes, solicitudes de productos a los proveedores, ventas, envíos de mercancía, control de inventarios, planificación y otros; entonces en los sistemas de agentes coordinativos se intentaría constituir diferentes tipos de agentes que deben asumir las funciones de las áreas antes mencionadas y coordinarse en función de lograr el objetivo de la organización. En este tipo de sistemas es de destacar además, que para cada tipo de agente, a pesar de tener un objetivo final común para el cual deben coordinarse, cada uno de ellos tiene una tarea específica que implica aptitudes y actitudes específicas en cuanto a, por ejemplo, capacidad de negociación, delegación, administración, cooperación o competencia que difieren de los otros agentes con los cuales han de coordinarse.
6. **Agentes Colaborativos:** son agentes que tienen la capacidad de comunicarse con otros agentes para posibilitar la cooperación y el intercambio de recursos y conocimientos. Generalmente cuando esto ocurre se habla de la constitución de un sistema dada una situación en donde un agente no es capaz de cumplir con la tarea que le ha sido asignada, debido a limitaciones funcionales con respecto a sus capacidades, por tanto este ha de solicitar ayuda a otros agentes que puedan colaborar al proveerle de recursos, conocimientos o ejecutar algunas actividades en su nombre y así permitirle acercarse al logro de su objetivo.

Los sistemas de agentes colaborativos y cooperativos han sido distinguidos como sistemas de agentes distribuidos, definidos como Sistemas Multi Agentes, que a su vez tienen sus propias especificaciones teóricas y técnicas que seguidamente serán ampliamente discutidas.

III.- SISTEMAS MULTI AGENTES

3.1.- Definición:

Para la Ingeniería de Software, la perspectiva que han aportado los agentes software ha representado una nueva vía para el análisis, diseño e implementación de sistemas de software complejos al ofrecer un poderoso repertorio de herramientas, técnicas y metáforas con el potencial para mejorar considerablemente las formas para definir e implementar muchos tipos de software (Jennings, Sycara & Wooldridge, 1998), por lo que está siendo cada vez mas requerida para el desarrollo de una amplia variedad de aplicaciones informáticas que van desde pequeños sistemas con tareas muy puntuales hasta sistemas complejos como los de comercio electrónico o de control de tráfico aéreo, por mencionar algunos. Sin embargo, esto también ha puesto en evidencia que no siempre es posible que un simple agente pueda resolver cualquier problema determinado, dado que tal posibilidad está condicionada por las capacidades del mismo y a su vez estas capacidades están limitadas por sus conocimientos, recursos técnicos y perspectivas (Sycara, 1998), que no todo el tiempo permiten alcanzar el objetivo que se le ha planteado al agente.

De manera sencilla los sistemas, sean estos poco o muy complejos, están definidos como un conjunto de elementos relacionados entre sí en función de lograr un objetivo. A partir de ello y vistas las consideraciones anteriores, resulta bastante sencillo entender que la idea de Sistemas Multi – Agentes (SMA), surge de una extrapolación del concepto de agentes hacia sistemas, donde intervengan diversos agentes software (elementos) que se relacionan al coordinarse entre sí, para intercambiar información o recursos que les sean de interés para el logro de sus objetivos. De este modo podemos adquirir una idea clara e intuitiva de lo que sería un sistema o sociedad de agentes múltiples, distribuidos y cooperativos que interaccionan entre sí (González, 2001)

Sin embargo, en los SMA no siempre existe de antemano un objetivo general para todo el sistema, tal como se definió en el párrafo anterior en el concepto de sistemas, como ocurre con los agentes colaborativos. En la mayoría de los casos, sin embargo, un sistema multi – agente, como un todo, persigue objetivos más complejos o amplios que la simple suma de las metas de los agentes que lo componen, como puede ser el caso de los agentes cooperativos.

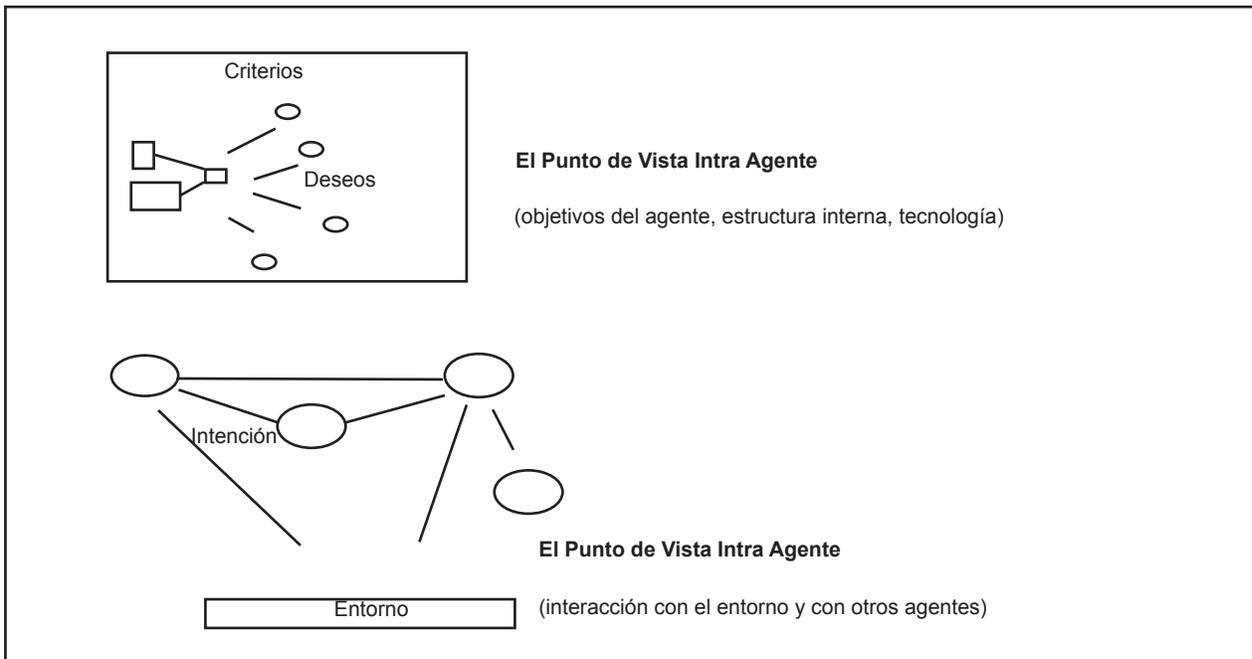


FIGURA 3.1. El punto de vista Intra – Agente vs. Inter – Agente (Zambonelli, Jennings, Ominici & Wooldridge; 2001)

La anterior perspectiva típicamente lleva a la concepción de un sistema multi agentes como una sociedad de agentes, donde interacciones mutuas entre agentes y con su ambiente conlleva a un comportamiento global óptimo. Correspondientemente, cada agente de un Sistema Multi – Agente puede ser visto de acuerdo a dos perspectivas diferentes (ver FIGURA 3.1): desde adentro, como un sistema (software) individual, con sus propios propósitos, estructuras y tecnologías (el punto de vista del intra – agente), y desde afuera, como una parte de una sociedad, interactuando con otros individuos, accediendo a recursos, y explotando la infraestructura social (el punto de vista Inter – agente). Estos últimos aspectos son los que hacen a los sistemas multi agente algo de una naturaleza radical con respecto a sistemas de software más tradicionales.

Desde el punto de vista conceptual, como características principales de los SMA pueden destacarse las siguientes (Jennings, Sycara & Wooldridge, 1998; Sycara, 1998]:

1. Cada agente posee información o capacidades incompletas para resolver el problema que se les plantea en conjunto, por tanto cada agente tiene una visión limitada.
2. Consecuentemente, cada agente individual del sistema tiene un conocimiento limitado del mismo, es decir, los agentes no disponen de información del sistema completo.

3. No existe un control total de todo el sistema porque los datos quedan descentralizados.
4. La computación es asíncrona en lo que respecta a la interacción entre los diversos elementos del SMA.
5. Deben permitir la interoperación con otros sistemas existentes, aunque ésta es de las tareas más complejas en la actualidad.

Por tanto, a pesar de las múltiples ventajas potenciales que ofrecen los Sistemas Multi - Agente, estos también enfrentan múltiples retos y dificultades. Uno de los aspectos que más preocupa a la hora de desarrollar SMA es el referido a las especificaciones para la interacción y comunicación entre agentes para lo que se han desarrollado Lenguajes de Comunicación de Agentes que serán ampliamente abordados en la sección siguiente.

3.2.- Lenguajes de Comunicación de Agentes:

El Lenguaje de Comunicación de Agentes (ACL por sus siglas en inglés) es el lenguaje que se ha desarrollado para permitir a los agentes comunicarse e intercambiar mensajes acerca de sus características, ambiente, requerimientos, creencias, necesidades, etc.

Los dos Lenguajes de Comunicación de Agentes más reconocidos son el KQML y el FIPA – ACL. El Knowledge Query and Manipulation Language (KQML), es quizás el más extendido de los dos y puede definirse como

un lenguaje y un protocolo de comunicación de nivel, orientado a mensajes para el intercambio de información, independientemente de la sintaxis del contenido y de la ontología aplicable (Labrou, Finin & Peng; 1999).

Por su parte el FIPA – ACL es un lenguaje desarrollado por la Foundation for Intelligent Physical Agents (FIPA), organización cuyo principal objetivo es la estructuración de especificaciones para maximizar la interoperabilidad entre los sistemas basados en agentes y cuyos alcances serán analizados más adelante.

En la mayoría de los aspectos sintácticos y operativos FIPA - ACL es casi igual a KQML (Hayzelden & Bigham, 1999). Su sintaxis es idéntica a la de KQML, con la excepción de algunos nombres diferentes para algunas palabras reservadas. Así mismo, al igual que KQML, este lenguaje no hace ningún tipo de compromiso previo con algún contenido particular de lenguaje, separando los lenguajes internos y externos.

Sin embargo, aunque estos lenguajes permiten resolver providencialmente los problemas para la comunicación necesarios para la constitución de SMA, aun quedan aspectos por resolver tales como la carencia de estándares para el desarrollo y operacionalización de agentes. En este sentido, el esfuerzo desarrollado por FIPA parece erigirse como el de mayor posibilidad para lograr la interoperabilidad entre agentes.

3.4.- FIPA como estándar para la interoperabilidad entre agentes:

La Foundation for Intelligent Physical Agents (FIPA) es una organización internacional sin fines de lucro, cuyo

propósito es producir estándares para agentes heterogéneos y sistemas basados en agentes (Foundation for Intelligent Physical Agents [FIPA], 2007). En el desarrollo de tales estándares, FIPA trabaja constantemente en la definición de un conjunto de especificaciones que van desde arquitecturas para soportar la comunicación entre agentes, lenguajes de comunicación y lenguajes de contenido para expresar tales mensajes y protocolos de interacción entre agentes. Todo ello está compilado en la especificación FIPA 97 y sus posteriores actualizaciones es la normativa donde se integran todos estos elementos que permiten el desarrollo de sistemas de agentes, de tal forma que se puedan crear sistemas complejos con un alto nivel de interoperabilidad, al posibilitar: la construcción y administración de un sistema de agentes compuesto por diferentes tipos de agentes, posiblemente construidos sobre diferentes plataformas entre si; la comunicación e interacción entre varios agentes para el logro de sus objetivos individuales o comunes; el uso por parte de los agentes software de sistemas de software no basado en agentes.

Este modelo de referencia se compone de un conjunto de entidades (ver FIGURA 3.3) que ofrecen diferentes servicios. Otras partes de la especificación son la interacción entre los humanos y los agentes, la gestión de la seguridad en los agentes, la gestión del soporte de la movilidad, el servicio de ontología y la guía del desarrollador. También se incluyen algunas implementaciones tales como entornos de desarrollo y operación de agentes (JADE y FIPA – OS) y algunas aplicaciones ejemplo como un asistente personal de viajes y un gerente de red, entre otros.

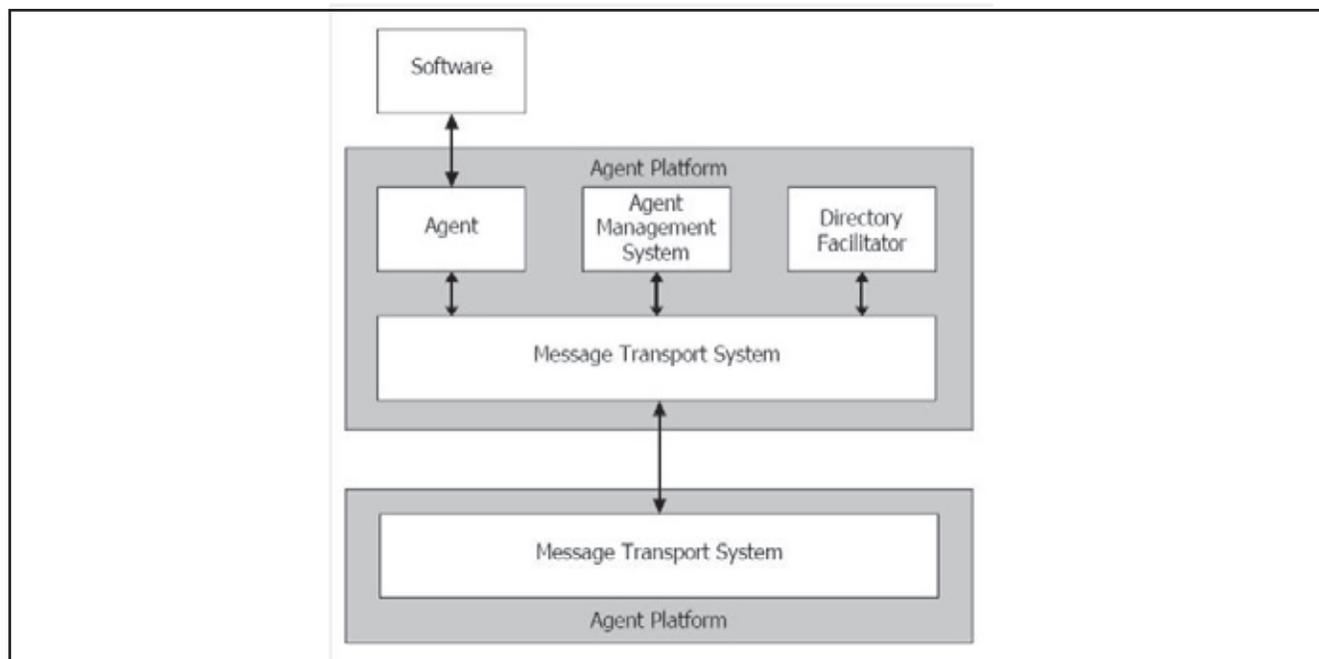


FIGURA 3.3. Modelo de referencia FIPA (FIPA, 2002)

Todos estos resultados han hecho que las especificaciones FIPA sean consideradas como una de las más importantes en el área de desarrollo de sistemas basados en agentes, dado que no solo intentan satisfacer los requisitos y problemas de desarrollo de Sistemas Multi agentes planteados con anterioridad, sino además por intentar plantear una arquitectura abierta y con estándares que permitan interoperabilidad entre agentes, independientemente de su plataforma de desarrollo. Pero a pesar de tales logros, estos aún están enfocados en garantizar la comunicación e inter operación entre agentes, por lo que ha sido muy poco el esfuerzo por lograr la incorporación de otras disciplinas que permitan el desarrollo de más amplias, nuevas y complejas aplicaciones aprovechando las ventajas que para ello ofrece la tecnología agentes. Ello no es tarea fácil, porque implica el manejo conceptual y empírico de elementos, características y restricciones que no siempre pueden acoplarse de manera sencilla. Ejemplo de ello es la incorporación de restricción de tiempo real a las tareas que un agente debe desarrollar, asunto que será estudiado a continuación, comenzando por una revisión de los conceptos y características asociadas a los Sistemas de Tiempo – Real.

IV.- SISTEMAS DE TIEMPO REAL

Un sistema computarizado de tiempo – real debe producir sus resultados dentro de un intervalo de tiempo específico (Thomas, Lambolais, Lesiour & Covadis; 1999). Esto quiere decir que para que el sistema sea aceptable no solo debe poseer corrección funcional (producir los resultados esperados de manera correcta), sino que debe poseer también corrección temporal, lo que es decir, que los resultados funcionalmente correctos deben producirse dentro de un intervalo de tiempo previamente determinado. De no producirse tales resultados dentro de este intervalo de tiempo, no solo afectarán la validez del sistema, sino que en algunos casos, harán al mismo totalmente inservible o inútil o lo que es lo mismo, una falla en el tiempo de respuesta puede considerarse tan mala como una respuesta incorrecta.

Una característica importante de un sistema de tiempo real es su criticidad, o lo que es lo mismo, el grado de tolerancia de fallas en el tiempo de respuestas y que permite clasificar a los STR en dos, a saber (Burns & Wellings, 2001; Thomas, Lambolais, Lesiour & Covadis, 1999):

1. Sistemas de Tiempo - Real Críticos (o duros): que poseen especificaciones de tiempo dentro de las cuales cada tarea debe producirse (iniciarse y culminarse) y que no admiten una sola respuesta fuera de tal plazo, porque de ser así, el sistema sería incorrecto.

2. Sistemas de Tiempo – Real Acríticos (o blandos): Son aquellos en los que se puede tolerar que se sobrepase algún plazo de tiempo de manera ocasional. Ello no implica que el tiempo de respuesta no sea importante, sino que el sistema puede ser aún válido aunque el tiempo de respuesta se salga de lo preestablecido de manera ocasional.

En un sistema con restricciones de tiempo real pueden coexistir partes con tolerancia a fallas de carácter críticas y acríticas. En todo caso los niveles de tolerancia a fallas en los tiempos de respuesta van a depender del tipo de ambiente en el cual el sistema se desenvuelve. En tal sentido, Kopetz (1991) considera dos tipos de ambientes:

1. Ambientes Estáticos: donde la frecuencia de los servicios requeridos al sistema están restringidos por un conocimiento a priori de las características del ambiente.
2. Ambientes Dinámicos: donde solamente es posible determinar los tipos de respuesta del ambiente ante situaciones específicas mediante modelos probabilísticos, dado que no existe información suficiente que permita predecir el comportamiento del ambiente ante estímulos o situaciones de incertidumbre.

Todas estas consideraciones tienden a dificultar el diseño y desarrollo de sistemas de tiempo real dado que para ello ha de tomarse en cuenta, además, la restricciones referentes a los tiempos en los que las acciones deben llevarse a cabo, los tiempos en que debe completarse cada acción, las respuestas a situaciones en las cuales no todos los tiempos se cumplen o a aquellas en las cuales los requisitos de tiempos cambian dinámicamente, que afecta desde la arquitectura del sistema, del hardware, los lenguajes e inclusive los Sistemas operativos.

A pesar de ello, hoy en día es bastante amplio el rango de metodologías de diseño e implementación de sistemas que toman en cuenta restricciones de tiempo real. Más sin embargo, en el campo de la integración de ello con los sistemas basados en agentes es muy poco lo que se ha hecho, tal como veremos a continuación.

4.4.- Agentes de Tiempo Real:

Varios autores coinciden en señalar que la integración de la tecnología de agentes con tiempo real puede permitir el desarrollo de aplicaciones que mejoren la flexibilidad y capacidad de respuesta de los sistemas de tiempo real (Hodys, 2001; Cingiser, Wolfe, Nair, Hodys & Uvarov, 2001; Bauer & Leake, 2001)

Si tomamos en cuenta las definiciones y características de los sistemas de tiempo real y de agentes podría-

mos decir que un agente de tiempo real es aquel agente que posee restricciones de tiempo para el desarrollo de sus tareas o responsabilidades (Cingiser, Wolfe, Nair, Hodys & Uvarov, 2001). Esto implica que el agente de tiempo real debe ser capaz de cumplir con las tareas que le han sido asignadas pero dentro de un período de tiempo específico, que puede ser crítico o no. Ahora bien si este agente con restricciones de tiempo real se encuentra formando parte de un SMA, es lógico pensar que todo el sistema se comportará como un Sistema Multi Agente De Tiempo Real (SMATR) dado que al haber tan siquiera uno solo de los agentes que conforman al sistema con tales características, la forma de accionar el sistema como tal estaría condicionada a que todos los agentes que conforma el mismo funcionen e interactúen de tal manera, que no solo cumplan sus tareas sino que la tarea de este o los varios agentes que posean restricciones de tiempo puedan hacerlo dentro de los tiempos requeridos, dado que lo contrario podría implicar la que este (estos) sean considerados inválidos, y así todo el sistema como tal.

Sin embargo, el desarrollo de agentes con restricciones de tiempo real no se presenta como una tarea fácil dado que, en principio:

1. La incorporación de tales restricciones puede poner en riesgo la autonomía del agente, al verse esta solapada por la necesidad de cumplir las tareas dentro del tiempo requerido
2. Aumenta la complejidad del agente dado que este tendría que poseer herramientas suficientes para poder adaptarse a ambientes altamente complejos y cambiantes.
3. No siempre podrá movilizarse hacia la mejor plataforma, dadas sus necesidades, sino hacia aquella que garantice que estas sean cubiertas en el tiempo requerido.
4. La habilidad de razonar y reaccionar del agente, podría aumentar de manera insoportable los tiempos de respuesta.

A su vez, en el caso de los sistemas multi agentes, esta problemática se torna aún más crítica dado que:

1. La interacción con otros agentes puede tornarse conflictiva, dada la necesidad de cumplir con los tiempos de respuesta especificados.
2. Dado que la capacidad de coordinación de los agentes exceptúa el caso en que el logro de los objetivos de uno de ellos implica la falla de todos los demás, se incrementaría el nivel de competitividad entre agentes, pudiendo producirse casos en los que inclusive algunos agentes rechacen colaborar con otros que tengan restricciones de tiempo.

3. Agrega una especie de plusvalía en cuanto a los conflictos que se presentan a la hora de asignar recursos y tiempos en los procesos de planificación basados en tiempo – real.

A pesar de las dificultades que presenta el desarrollo de SMATR, existen algunos esfuerzos que han intentado abordar el desarrollo de SMA que manejen especificaciones de tiempo real, que en general tienen en común los siguientes (Soler, Julian, Rebollo, Carrasosa & Botti, 2002; Cingiser, Wolfe, Nair, Hodys & Uvarov, 2001; Hodys, 2001; Bauer & Leake, 2001; Surka, Brito & Harvey, 2001; Weiss & Steger, 1998):

1. Se enfocan básicamente en el desarrollo de SMA en TR a nivel de conceptualización y de arquitecturas, con muy pocas especificaciones acerca de la comunicación, interacción y, en general la operatividad de los agentes.
2. Las restricciones de tiempo son en su totalidad blandas, no se toman en cuenta agentes con restricciones de tiempo críticas

Estas observaciones no hacen más que confirmar que el desarrollo de SMATR es un área que apenas está en un muy temprano desarrollo pero que sin embargo muestra un creciente interés por la misma que solo puede entenderse dada la potencialidad que esta representa para el desarrollo e aplicaciones software con mayores y mejores alcances y utilidades.

Para abordar tal tarea es importante que se empiece por desarrollar una arquitectura que permita definir un patrón para el desarrollo de SMATR. Si tomamos en cuenta que una arquitectura es una estructura que permite definir los componentes de software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos (Pressman, 2001), es obvio asumir que el desarrollo de una tecnología debe comenzar por la construcción de una arquitectura que permita constituir un marco de referencia acerca de cómo se estructuran, en nuestro caso, los agentes que han de interactuar tomando en cuenta restricciones de tiempo real.

Las Arquitecturas de Sistemas de Agentes analizan los agentes como entidades interactivas que proporcionan y consumen servicios (Huns & Singh, 1998). Por tanto facilitan la descripción de las operaciones y las interacciones de los agentes bajo las restricciones del entorno, y les permiten aprovechar las facilidades y los servicios disponibles. No obstante, la mayoría de las investigaciones realizadas en el campo de los sistemas multi agente se centran en diseñar arquitecturas que permitan esquemas de negociación complejos y resolución de tareas de alto nivel, y no en proporcionar sistemas que se adapten a unas determinadas restricciones temporales (Marsá, Sánchez & Velazco, 2004).

Para Julian, Carrascosa, Rebollo, Soler & Botti (2002), el uso de Sistemas Multi Agentes de Tiempo Real tiene sentido dentro de entornos con restricciones temporales críticas, donde el sistema pueda ser controlado por agentes autónomos, y donde dichos agentes necesiten comunicarse entre sí para mejorar el grado de cumplimiento de los objetivos del sistema. En este tipo de entornos, los agentes necesitan actuar de forma autónoma, pero avanzando hacia un objetivo común. Además, es necesario garantizar la respuesta del sistema en tiempo real, y debe evitarse el establecimiento de comunicaciones excesivamente voluminosas entre agentes.

De acuerdo con Marsá, Sánchez & Velazco (2004), las diferentes plataformas y arquitecturas planteadas hasta ahora para la construcción de Sistemas Multi Agentes de Tiempo Real (SMATR) cumplen en mayor o menor grado cada uno de estos requisitos, y los afrontan desde diferentes estrategias. Visto así, pareciera que entonces estos requisitos son suficientes para desarrollar SMATR, cualquiera sea su naturaleza. Sin embargo, al ser revisados más detenidamente los mismos parecieran ser suficientes solo para el caso de SMATR en los que de antemano se ha definido que el objetivo o la meta que ha de satisfacer el sistema como un todo debe alcanzarse atendiendo a ciertas restricciones de tiempo, independientemente de que los agentes que conforman el sistema deban atender de manera individual algún tipo de restricción temporal. En este caso, las actividades y tareas a desarrollar por cada agente pueden ser previamente acotadas. Consecuentemente, es responsabilidad del diseñador acotar los tiempos de respuesta de las tareas de cada agente para garantizar que los tiempos de respuesta de las mismas no entren en conflicto con la corrección temporal del sistema general. Este tipo de arquitecturas pareciera entonces adaptarse más a aquellos SMA compuestos por agentes de tipo coordinativo, que interactúan en función de una distribución de tareas dentro un entorno casi siempre predecible y para el logro de un objetivo superior a la tarea específica que desarrolla cada uno.

A su vez, ese tipo de arquitecturas no considera aquellos casos en los cuales la interacción entre los agentes es de carácter colaborativo. Es decir, cuando estos cooperan entre sí para compartir conocimientos y/o recursos en función de cumplir con la tarea asignada en vista de que sus propias capacidades le son insuficientes. En estos casos no existe un objetivo del sistema previamente establecido y definido por un diseñador o desarrollador, dado que el mismo se constituye una vez que se establece la comunicación entre agentes en aras de colaborar mutuamente. Por tanto, no es posible prever y acotar los términos de la interacción entre los diferentes agentes, lo que si es claro es que cuando uno de los que interviene o forma parte del proceso tiene restricciones de tiempo, este hace que toda la actividad de los otros agentes este delimitada también por tales restricciones, convirtiendo a todo el sistema multi agente en un Sistema Multi Agente de Tiempo Real.

Es en este tipo de ocasiones en que la interacción en los Sistemas Multi Agentes puede tornarse conflictiva, sobre todo porque ello implica agregar restricciones de tiempo, y por tanto de uso y disfrute de recursos a agentes que de antemano no tendrían tales, y quienes por tanto, podrían preferir relacionarse con otros que presentan menores "exigencias" a la hora de colaborar. Para evitar tales conflictos, es necesario desarrollar Arquitecturas de Sistemas Multi Agentes de Tiempo Real Cooperativos, que incluyan mecanismos de negociación para la resolución de conflictos entre agentes con restricciones de Tiempo Real y otros agentes.

Para lograr ello se sugiere la incorporación de un agente negociador que actuaría en el caso de que un agente de Tiempo Real necesite interactuar con otros agentes para que estos colaboren en el logro de sus metas. Para establecer esta colaboración el *Agente Negociador de Tiempo Real* sería quien buscaría los posibles agentes que por sus capacidades pueden contribuir a lograr la meta del Agente de Tiempo Real en necesidad de colaboración (Figura 5.1).



Figura 5.1. Proceso de Colaboración entre agentes para Arquitecturas de SMATR Cooperativos

Este negociador sería una especie de intermediario para el logro de convenios de cooperación sin que entren en conflicto los objetivos de cada una de las partes con respecto a los tiempos en los cuales estos deben ser alcanzados.

Adicionalmente, una vez establecidos tales acuerdos, en algunos de los casos y en función de los términos de los mismos, podrían considerarse la incorporación al Sistemas Multi Agentes de Tiempo Real Cooperativos, de un *Agente Planificador* con la responsabilidad de planificar los tiempos y términos de la interacción entre los agentes con el fin de satisfacer las necesidades individuales de cada uno y las restricciones de tiempo que se impongan. Las gestiones de este *Agente Planificador* se hacen más críticas en la medida en que los agentes que interactúan incorporan más restricciones de tiempo.

Tanto el *Agente Negociador de Tiempo Real* como el *Agente Planificador* son agentes de tiempo real, cuyas restricciones están impuestas por los entes a los cuales representan.

VI.- CONCLUSIONES Y TRABAJO A FUTURO

Aunque existen diferentes líneas de investigación que pretenden aplicar la tecnología de sistemas multi agente a sistemas de tiempo real, tanto críticos como acríticos, la mayoría de ellas se orientan a la construcción de implementaciones en funcionamiento para determinadas aplicaciones. Así mismo, las pocas propuestas de arquitecturas que existen se basan en un enfoque que otorga prioridad a Sistemas Multi Agentes de naturaleza coordinativa, dejando de lado aquellos casos en los que los agentes se reúnen para colaborar entre sí para el logro de sus objetivos particulares, sin que se haya definido de antemano la dinámica de la interacción entre ellos.

En tal sentido se advierte la necesidad de abocarse hacia el desarrollo de Arquitecturas de Sistemas Multi Agentes de Tiempo Real Cooperativos, para lo que se asoman algunos elementos iniciales a tomar en cuentas tales como la consideración de la existencia de lo que hemos denominado un *Agente Negociador de Tiempo Real* así como el *Agente Planificador*.

Estos dos actores deben ser desarrollados en lo interno de manera más detallada en futuros trabajos. Así mismo, tal arquitectura debería contener desde el principio la intención de permitir interoperabilidad entre diferentes tipos de agentes sean estos de tiempo real o no con un carácter estándar, alejándose de visiones específicas, al menos de antemano.

En este sentido para la inclusión de este último elemento podría tomarse en cuenta el trabajo adelantado por la arquitectura FIPA, dado que, aunque no inclu-

ye aun prestaciones de tiempo real, si ha reconocido la importancia de las mismas para los sistemas multi agentes y posee la ventaja de partir con la intención de ser estándar, es decir, servir para la interacción de cualquier tipo de agentes independientemente de la plataforma original de desarrollo de los mismos.

VII. - REFERENCIAS BIBLIOGRÁFICAS

- Agent Working Group. 2000. Agent Technology Green Paper. Object Magnagement Group, Needham (Massachusetts).
- Amandi, A. 2001. Desarrollo de Sistemas Multi - Agentes. Revista Iberoamericana de Inteligencia Artificial. 5(13): 33-35.
- Bauer, T & Leake, D. 2001. A Research Agent Architecture for Real time data Collection and Analysis. Working Notes of the Second International Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. Fifth International Conference on Intelligent Agents, Montreal: 61-66.
- Burns A. & Wellings A. 2001. Real-Time Systems and Programming Languages Ada 95: Real-Time Java and Real-Time POSIX. Third Edition. Addison Wesley Longmain, USA.
- Cingiser, L., Wolfe, V., Nair, L., Hodys, E. & Uvarov, O. 2001. A Real-Time Multi-Agent System Architecture for E-Commerce Applications. ISADS 2001: 357-364
- Foundation for Intelligent Physical Agents. 2007. FIPA's Mission Statement [En línea]. Disponible en: www.fipa.org. Revisado 12/01/06
- Foundation for Intelligent Physical Agents. 2002. FIPA Abstract Architecture Specification [En línea]. Disponible en: <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>. Revisado 18/10/05
- Franklin S. & Graesner A. 1996. Is it an Agent or Just a Program: A Taxonomy for Autonomous Agents. En: Intelligent Agents 3: Agent Theories, Architectures, and Languages. Proceedings of the ICAI '96 Workshop, Budapest, Hungary, 12-13 August 1996. New York: Springer-Verlag.
- González, J. 2001. Protocolo activo para transmisiones garantizadas sobre una arquitectura distribuida y multiagente en redes ATM. Tesis Doctoral, Universidad Politécnica de Cataluña, Barcelona.
- Griss, M.L. 2000. My Agent Will Call Your Agent. www.sdmagazine.com. Febrero
- Hayes-Roth, B. 1995. An Architecture for Adaptative Intelligent Systems. Artificial Intelligence: Special Issue on Agents and Interactivity. 72(1-2):329-365.

- Hayzelden, A. & Bigham, J. 1999. Agent Technology in communication Systems: An Overview. *The Knowledge Engineering Review*. 14(4): 341-375.
- Hodys E. 2001. A Scheduling Algorithm for a Real-Time Multi-Agent System. Tesis de Maestría. University of Rodhe Island. Rodhe Island, USA
- Huns, M. N. & Singh, M. P. 1998. Agents and Multi Agents Systems: Themes, Approaches and Challenges. En: Huns, M. N. & Singh, M. P. (Editores). *Readings in Agents*. Morgan Kauffman Publishers, San Francisco (USA).
- Jennings, N., Sycara, K. & Wooldridge, M. 1998. A Roadmap of Agent Research and Development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7-38.
- Jennings, N. & Wooldridge, M. (Ed.). 1998. *Agent technology – Foundations, Applications and Markets*. Springer – UNICOM, Londres.
- Julian, V., Carrascosa, C., Rebollo, M., Soler, J., Botti, V. 2002. Simba: an approach for real-time multi-agent systems. In: *Proceedings of V Conferencia Catalana d'Intelligencia Articial, Castelló*, Springer-Verlag: 282-293.
- Kopetz, H. 1997. *Real Time Systems, Design Principles for Distributed Embedded Applications*. Boston. Kluwer Academic Publishers, Boston.
- Labrou, Y., Finin, T. & Peng, Y. 1999. Agent Communication Languages: The Current Landscape. *IEEE Intelligent Systems*. 14(2):45-52.
- Maes, P. 1994. Modeling Adaptative Autonomous Agents. *Artificial Life Journal*, MIT Press. 1(1-2):135-162.
- Marsá I., Sánchez, S. & Velasco J. R. 2004. Sistemas Multiagente de Tiempo Real. I Congreso Iberoamericano sobre Computación Ubícua (CICU 2005), España: 1-8.
- Nwana, H. 1996. Software Agents: An Overview. *Knowledge Engineering Review*. Cambridge University Press. 11 (3): 1-40.
- Piszc, A. 1988. A Brief Overview of Software Agent Technology. MITRE Corp, McLean (Virginia).
- Pressman, R. S. 2001. *Ingeniería de Software: Un enfoque Práctico*. 5a Edición, Mc Graw Hill, Madrid.
- Russell, S y Norvig, P. 1995. *Artificial Intelligence: A modern Approach*. 2a Edición, Prentice Hall, New Jersey.
- Shoham Y. 1999. What we talk About when we talk about software agents. *IEEE Intelligent Systems*. 14 (2):28-31.
- Smith, D.C., Cypher, A. & Spoher, J. 1994. KidSim: Programming Agents Without a Programming Language. *Communications of the ACM*. 37(7):54-67.
- Soler, J., Julian, V., Rebollo, M., Carrascosa, C. & Botti, V. 2002. Towards a Real-Time Multi-Agent System Architecture. 1st International Workshop on Challenges in Open Agent Systems, Bologna, Italy: 1-12.
- Surka, D., Brito, M. & Harvey, C. 2001. The Real-Time ObjectAgent Software Architecture for Distributed Satellite Systems. 2001 IEEE Aerospace Conference Proceedings, Big Sky, Montana.
- Sycara, K. 1998. Multiagent Systems. *AI Magazine*. 19(2):79–92.
- Tolosa G. & Bordignon F. 1999. Revisión: tecnología de agentes de software. *Ciencia e Informatica*. 28(3):302-309.
- Thomas, L., Lambolais, T., Lesiour, R., Covadis, A. 1999. Architectural Techniques for the Description and Validation of Distributed Real Time Systems. *Object – Oriented Real-Time Distributed Computing*. 1999. (ISORC '99). *Proceedings 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, Saint – Malo, France: 323 – 331.
- Universidad Santa María. 2000. *Normas para la Elaboración, Presentación y Evaluación de los Trabajos Especiales de Grado*. Autor. Caracas.
- Weiss, R. & Steger, C. 1998. Design and implementation of a real-time multi-agent system. 9th Mediterranean Electrotechnical Conference, melecon '98, Tel-Aviv, Israel: 1269-1273.
- Zambonelli F, Jennings N, Ominici A & Wooldridge M. 2001. Agent – Oriented Software Engineering for Internet Applications. En: Ominici, A., Zambonelli, F., Klusch, M. & Tolksdorf, R. (Editores). *Coordination of Internet Agents*. Spring Verlag. pp. 326 – 346